Fine-tuning BERT for Amazon Review Rating Prediction

- Leveraging BERT-base-cased on Amazon Reviews Dataset

Team Members:

- Bhanu Teja Nandina
- Krishna Vamsi Nadh Arikatla
- Likitha Y

11/28/2023

Project Overview:

- Fine-tuning BERT for Amazon review rating prediction.
- To develop a model capable of predicting review ratings on Amazon based on the review text using BERT.
- Statement of Value - Why is this project worth doing?
  - Accurate sentiment analysis in reviews is crucial for businesses and consumers.
- Predicting review ratings provides valuable insights for product improvement and customer satisfaction.

Dataset

- Amazon Reviews: https://cseweb.ucsd.edu/~jmcauley/datasets.html#amazon_reviews

Overview:

- A diverse dataset containing product reviews from Amazon.

Large and representative, encompassing various product categories.

Preprocessing:

- Cleaned and normalized to ensure uniformity.
- Tokenization and encoding applied to convert raw text into suitable input for BERT.
- We have used BERT-base tokenizer

```
tokenizer = BertTokenizer.from_pretrained('bert-base-cased')
```

Model:

- Pre-trained Model: BERT-base-cased

Overview:

- BERT-base-cased is a pre-trained model with case-sensitive information.
- Trained on massive corpora, providing a strong foundation for understanding language context.

Fine-tuning:

- Specific layers of BERT are fine-tuned for the task of predicting Amazon review ratings.
- Last layers has changed from classification to regression layer. To predict the numerical rating.

Tools:

- Hugging Face Transformers, PyTorch

Hugging Face Transformers:

- Open-source library providing easy access to pre-trained models like BERT.
- Facilitates fine-tuning and integration into custom applications.

PyTorch:

- PyTorch used as the underlying deep learning framework.
- Provides the computational backbone for implementing and training the BERT-based model

Training and Evaluation

Training:

- BERT is fine-tuned to the specific task of predicting review ratings on the Amazon Reviews dataset.
- Model is trained for 3 epochs using batch size of 32.
- Model took 3 hours to train on google colab.

```
Epoch 1/3: 100%|██████████| 1585/1585 [34:42<00:00,  1.31s/batch, Loss=0.813]
Epoch 1/3, Average Loss: 0.8128
Epoch 2/3: 100%|██████████| 1585/1585 [34:37<00:00,  1.31s/batch, Loss=0.315]
Epoch 2/3, Average Loss: 0.3147
Epoch 3/3: 100%|██████████| 1585/1585 [34:39<00:00,  1.31s/batch, Loss=0.233]
Epoch 3/3, Average Loss: 0.2332
```

Evaluation:

•Mean Squared Error (MSE) is employed as the primary metric to evaluate the model's performance.

•MSE measures the average squared difference between predicted and actual review ratings.

- We have got test MSE of **0.23**.

```
Mean Squared Error: 0.1929172469279199
```

Deploment:

- We have used Flask for deployment
- Use can be able to enter review and he get's the rating of the review displayed on the page.